December 2020

# PROCEDURALLY GENERATING ARTIFICIAL PHOTOS OF TEXT DOCUMENTS IN VARIOUS BACKGROUNDS

HP INC

## Title

Procedurally Generating Artificial Photos of Text Documents in Various Backgrounds

## Abstract

Text document photos remain a highly important resource in the development or training of document segmentation and Optical Character Recognition (OCR) applications. This is due to the quality assessment of algorithms for these tasks is performed using unknown images, which guarantees their robustness, and that Machine Learning techniques need large datasets with a great number of samples with high inter and intraclass diversity. This work presents a method to procedurally generate simulations of photos of text documents with different backgrounds. Using Digital Image Processing techniques, this process eliminates the need of creating a real dataset by hand and increases the variability of the available data. The dataset generated can have any number of samples, can be used in models that need smaller or larger datasets.

## Description

We use a background image and an image of a text document to procedurally create a simulation of a photo taken from a camera. All the steps are summarized in Figure 1.

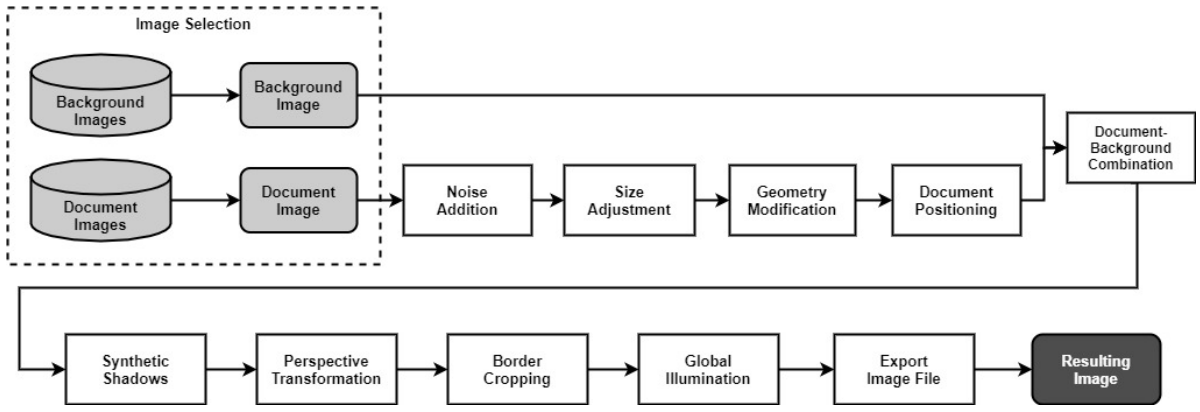


Figure 1. Pipeline of the simulated photo document image generation.

At the beginning of the process, there are a set of background images and a set of document images. These datasets are passed as input at the start of the pipeline. Some examples of background images are presented in Figure 2. Examples of document images are presented in Figure 3. Note that there is no restriction in the image appearances.

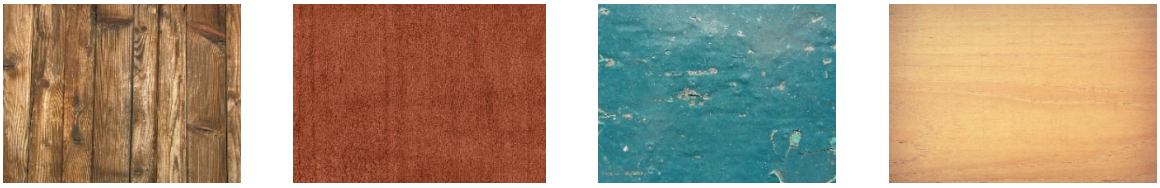

Figure 2. Examples of background images to be used in the generation process.

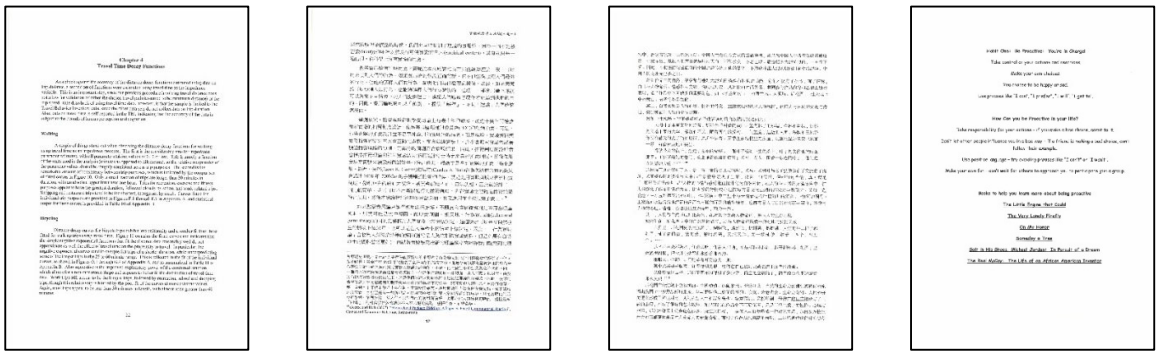

Figure 3. Examples of document images to be used in the generation process.

The backgrounds are put in a folder and the user can opt to use one specific background or to let the choice be made randomly by the algorithm. Similarly, the user can choose a document or pass a folder name to the code, which will randomly choose a document image. This process occurs once to generate a single image. If the user wants more than one image, the process can be executed multiples times in sequence.

We then apply some noise only to the document to better simulate real photography. One example of distortion is blurring. If this distortion is chosen, the original full-resolution document image receives a blur change. The intensity of the distortion is controlled by parameters that select randomly between a minimum and a maximum value. Note that both minimum and maximum values can be controlled by the user. The intensity of the distortion will affect the clarity of the text characters. Thus, if the value is high, a recognition algorithm applied in the result may fail to correctly identify the characters. An example of blur distortion is shown in Figure 4.

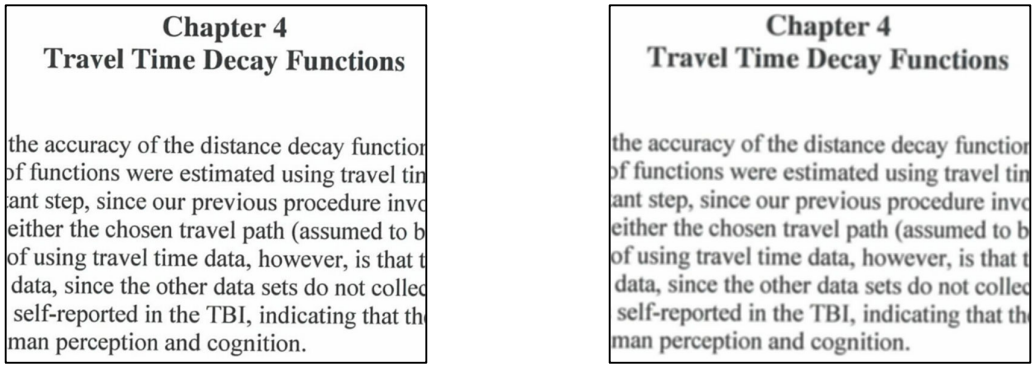


Figure 4. Zoomed view of a blur distortion. Left: Original image. Right: Blurred counterpart.

There are some properties that must be met to correctly generate a plausible result. One of them is that the document should be fully inside the image. If the document exceeds the image borders, part of its content could be outside the image. Also, if a corner is not present in the image, a recognition algorithm that expects that all four corners are visible may fail. To ensure that the document has the appropriate size, it is resized according to the width and height of the background. Note that this process can also grow the size of the document if it is too small in relation to the background. An example of the result after resizing the original document is shown in Figure 5.

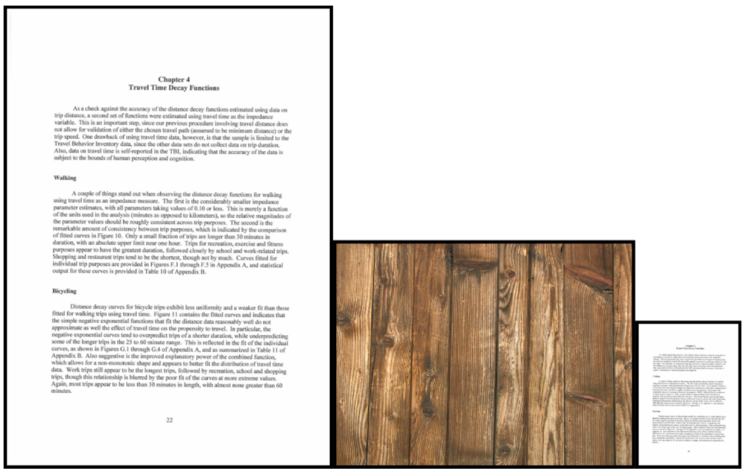

Figure 5. The difference in the size of the original document in relation to the background. Left: Original size. Center: Background. Right: Document size after the resize operation.

Another type of possible document modification is to change its orientation (Figure 6). There are three main types of possible rotations, described as follows.

a) No rotation: the document is kept in the same orientation.
b) Restrict rotation: the user can define a minimum and a maximum rotation angle. The effective angle will be chosen randomly between these values.
c) Free rotation: the angle will be randomly chosen between any value from 0° to 360°. This option gives full control of the rotation, but it can generate upside-down images.

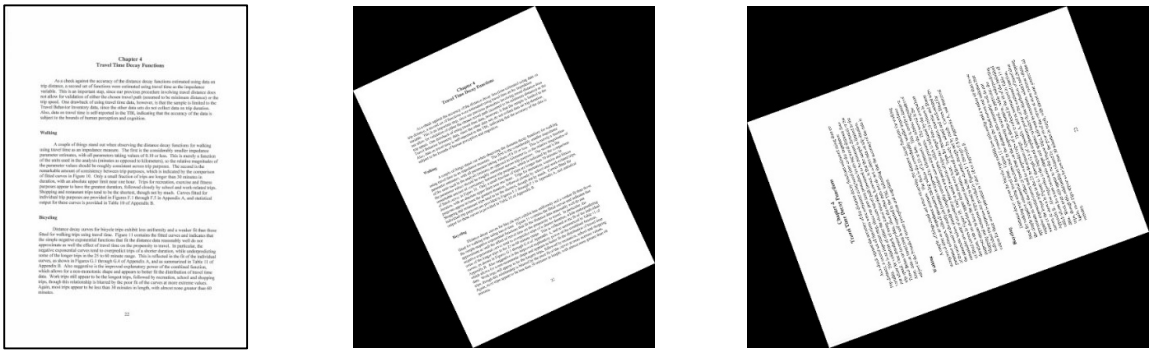

Figure 6. Orientation changes. Left: No rotation. Center: 25° rotation. Right: 110° rotation.

In its simplest case, the image is placed in the center of the background (Figure 7a). However, this would limit the variation of the documents and would not be proper to evaluate segmentation algorithms, since they could bias their answer towards the center of the image. To provide a more diverse generation, the document position in relation to the background can also be modified. This allows the generation of images with documents in a variety of positions (Figure 7b).

Figure 7. Two types of positioning. Left: Centered document. Right: Translated document.

Nevertheless, not all translations should be allowed. For instance, the document could not have a part outside the background border. Or if it is too close to the border, the corners would possibly end up outside of the image after subsequential adjustments. These problems are shown in Figure 8.

Figure 8. Inappropriate positioning. Left: Outside of the border. Right: Too close to the border.

To ensure good placement in relation to the background, there is a defined potential position space. It is defined as the remaining space after ignoring a margin, either given by the user or randomly chosen. For example, if the document position is restricted, it should not be placed inside the reddish space shown in Figure 9.

Figure 9. No part of the document should be inside the red border.

After defining the position of the document in relation to the background, the next step is to combine the document and the background. The result is shown in Figure 10.

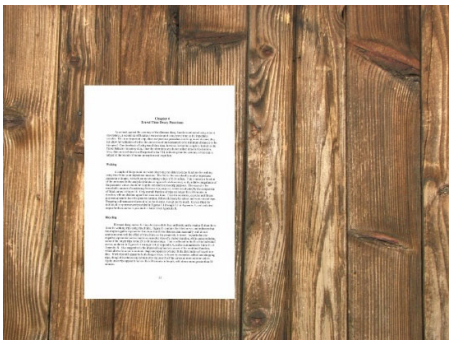

Figure 10. Document and background are combined in a single image.

The next step is to create simulations of shadows. This is achieved by applying a semi-transparent shape in arbitrary places of the document. Some examples of different shapes of shadows applied in the result of the previous step are presented below.

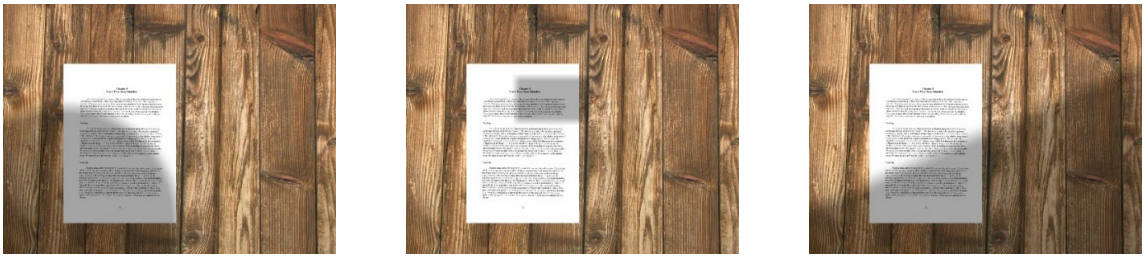

Figure 11. Three examples of shadows. Left: Circular. Center: Rectangular. Right: Entire image.

Since the goal is to simulate a photo, there are at most two possible perspective variations, because the camera is a two-dimensional plane. To simulate these possible variations, we apply two

perspective transformations along with one of the 2D axes. Note also that the points are not freely movable, they must follow the same pattern following a plane rotation. Thus, there is a total of four possible perspective variations, which are presented in Figure 12. With a combination of two of the four distortions, we can simulate all possible camera perspective adjustments.

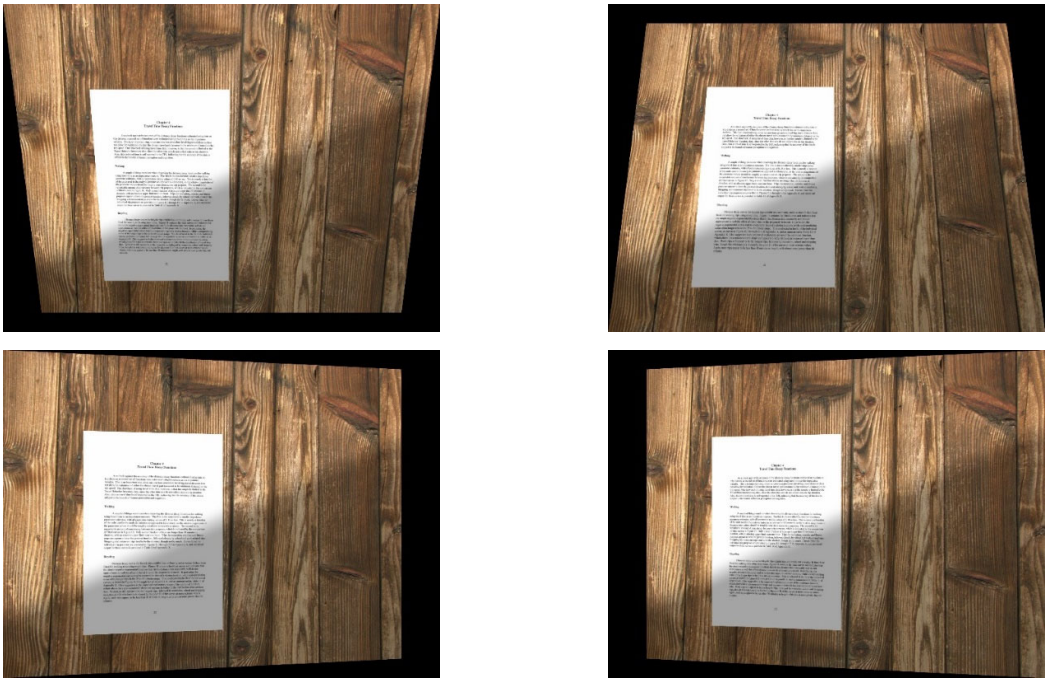

Figure 12. All possible single perspective distortion.

After the perspective adjustments, the images end up with black pixels where there were image pixels previously. This is undesired since we do not expect black parts in a real photo. To fix this problem, we crop the borders of the images according to the perspective distortion applied. In the end, the image has no more black pixels, as can be seen in Figure 13.

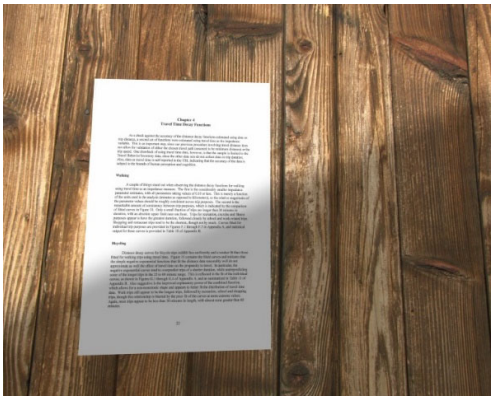

Figure 13. Result after cropping the unwanted borders.

Often, the illumination of a real photo implies images that end up being darker than their real counterparts. Since this process works with digital images, the results can be brighter than expected, which may not represent the illumination of real photos. To better represent them as possible real photos, we can decrease the global illumination of the full image, according to a light parameter. An example can be seen in Figure 14.

Figure 14. Global illumination adjustment. Left: Original image. Right: Reduced illumination.

With the result of the last step, the image is ready to be exported. The image format can be chosen from a great variety of options, including BMP, JPEG, and PNG. All intermediate results from the given document and background are presented below.

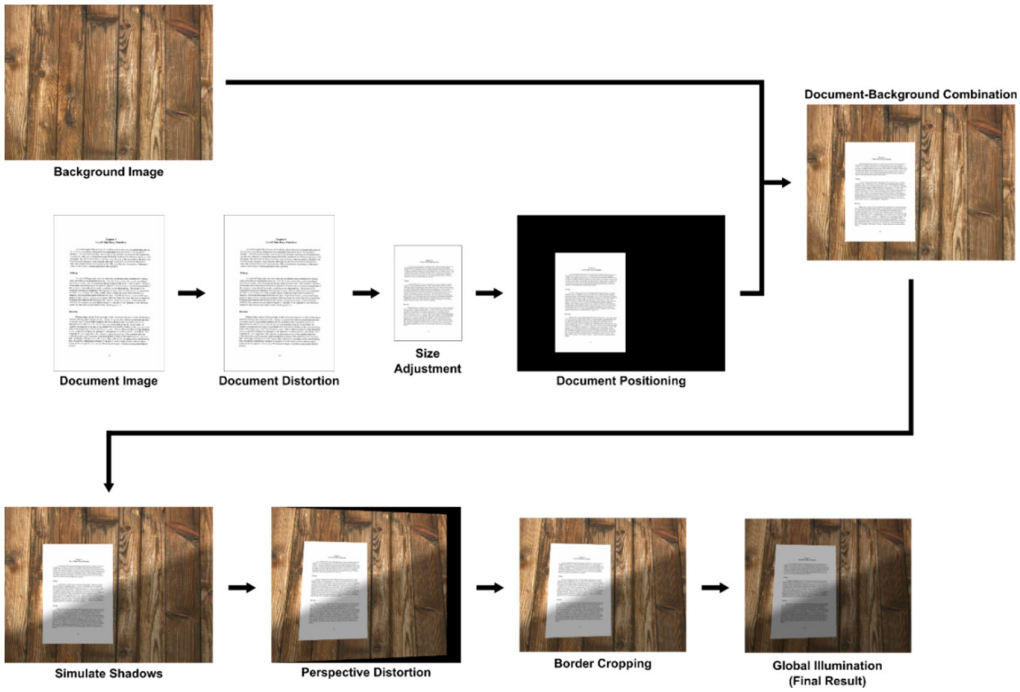


Figure 15. All intermediate steps of an image generation using the pipeline presented.

*Disclosed by Paulo Serafim, Valberto Enoc, Caio Lucas, Instituto Atlântico and Lucio Cossio, Joan Alminhana, HP Inc.*